

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): FUJITA et al.

Application No.: 10/022,794

Filed: December 20, 2001

Title: MICROCOMPUTER

Attorney Docket No.: 01-242

Group Art Unit:

Examiner:

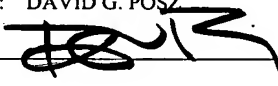


March 11, 2002

**CERTIFICATE OF HAND DELIVERY**

I hereby certify that this correspondence is being hand delivered to and deposited with the USPTO at Customer Service Window, Office of Initial Patent Examination, Crystal Plaza, Building 2, Room 1B03, 2011 South Clark Place, Arlington, VA 22202 on March 11, 2002.

Typed Name: DAVID G. POSZ

Signature: 

**SUBMISSION OF PRIORITY CLAIM AND PRIORITY DOCUMENTS**

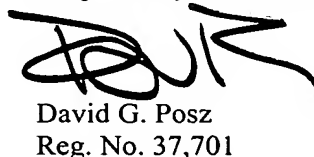
Assistant Commissioner for Patents  
Washington, D.C. 20231

Dear Sir:

Pursuant to the provisions of 35 U.S.C. § 119, it is respectfully requested that the present application be given the benefit of the foreign filing date of the following foreign application(s). A certified copy of each application is enclosed.

<u>Application Number</u>	<u>Country</u>	<u>Filing Date</u>
2000-402422	JAPAN	December 28, 2000
2000-402423	JAPAN	December 28, 2000
2000-402425	JAPAN	December 28, 2000

Respectfully submitted,

  
David G. Posz  
Reg. No. 37,701

Law Offices of David G. Posz  
2000 L Street, N.W.  
Suite 200  
Washington, D.C. 20036  
(202) 416-1638  
Customer No. 23400

日 本 国 特 許 庁  
JAPAN PATENT OFFICE



58696-03  
KR/st  
1-242

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日  
Date of Application:

2000年12月28日

出 願 番 号  
Application Number:

特願2000-402422

[ ST.10/C ]:

[ JP 2000-402422 ]

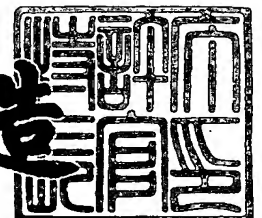
出 願 人  
Applicant(s):

株式会社デンソー

2002年 1月11日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3114216

【書類名】 特許願

【整理番号】 PNID3666

【提出日】 平成12年12月28日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 7/00

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

    【氏名】 藤田 洋一

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

    【氏名】 新田 修一

【発明者】

    【住所又は居所】 愛知県刈谷市昭和町1丁目1番地 株式会社デンソー内

    【氏名】 妹尾 伸一

【特許出願人】

    【識別番号】 000004260

    【氏名又は名称】 株式会社デンソー

【代理人】

    【識別番号】 100082500

    【弁理士】

    【氏名又は名称】 足立 勉

    【電話番号】 052-231-7835

【手数料の表示】

    【予納台帳番号】 007102

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

特 2 0 0 0 - 4 0 2 4 2 2

【包括委任状番号】 9004766

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 マイクロコンピュータ

【特許請求の範囲】

【請求項 1】 プログラムに従い動作する CPU と、

該 CPU を動作させるためのメインクロックを生成するメインクロック生成手段と

前記メインクロックよりも周波数が低いサブクロックを受けて動作すると共に、前記 CPU を間欠的に動作させるための制御を行う間欠動作制御手段と、

を備え、前記 CPU が、動作を停止する際に前記間欠動作制御手段へ停止指令を出力し、前記間欠動作制御手段が、前記停止指令を受けると、前記メインクロック生成手段の動作を停止させると共に、所定の設定時間の計時を開始し、その設定時間の経過後に、前記メインクロック生成手段の動作を再開させて前記 CPU を停止状態から動作状態へと起床させることにより、前記 CPU の間欠動作が実施されるマイクロコンピュータであって、

前記サブクロックを受けて動作して、前記 CPU が停止状態である期間を計測すると共に、その計測値が前記 CPU によって読み取り可能な間欠時間計測手段を備えていること、

を特徴とするマイクロコンピュータ。

【請求項 2】 請求項 1 に記載のマイクロコンピュータにおいて、

前記間欠時間計測手段は、

前記 CPU からの動作モード切替指令によって、前記 CPU が停止状態である期間を計測する第 1 の動作モードと、時間を継続して計測する第 2 の動作モードとの何れかの動作モードに設定されると共に、少なくとも、動作モードが前記第 2 の動作モードに設定されている場合には、前記 CPU からのクリア指令によって計測値のクリアが行えるように構成されていること、

を特徴とするマイクロコンピュータ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マイクロコンピュータに関し、特に、予定の動作を所定時間が経過してから実施する電子制御装置に好適なマイクロコンピュータに関する。

【0002】

【従来の技術】

従来より、電子制御装置においては、例えば、通信相手の装置へ起動信号を送信してから所定時間が経過した後で、その通信相手との通信動作を実施したり、外部のスイッチ等からの信号がアクティブレベルに変化したことを検知してから所定時間が経過した後で、ランプやブザー等の電気負荷を駆動するための動作を実施する、といった具合に、予定の動作を所定時間が経過してから実施するように構成されることがある。尚、以下、このような機能（即ち、予定の動作を所定時間が経過してから実施する、という機能）を「タイマ機能」という。

【0003】

また、例えば車両に搭載される電子制御装置では、バッテリーへの充電が行われないエンジン停止時に上記のタイマ機能を実現しなければならない場合がある。このため、より少ない消費電力でタイマ機能を実現する必要がある。

そして、従来の電子制御装置では、上記のタイマ機能を少ない消費電流（消費電力）で実現するために、下記の〔従来例1〕～〔従来例3〕の何れかの方法を採用していた。

【0004】

〔従来例1〕

まず、図4（a）に示すように、電子制御装置の動作を制御するマイクロコンピュータ（以下、マイコンという）101の外部に、タイマ回路103を設ける。

【0005】

そして、この電子制御装置において、マイコン101は、実行すべき処理が無く、動作しなくても良い状態になったと判断すると、図4（b）に示すように、当該マイコン101の内部で動作クロックを発生させている発振回路（図示省略）の動作を停止させて自分の動作を停止すると共に、タイマ回路103へ動作指示を出す。

## 【0006】

すると、タイマ回路103は、計時を開始して、所定のタイマ時間が経過した時に、マイコン101を停止状態から動作状態へと起床（ウェイクアップ）させるためのウェイクアップ信号を出力する。尚、タイマ回路103は、マイコン101の動作クロックよりも低い周波数のクロックであって、当該タイマ回路専用の発振回路105で発生されるクロックによって動作する。

## 【0007】

そして、マイコン103は、タイマ回路103からのウェイクアップ信号によって停止状態から動作状態へと起床し、予定の動作に該当する処理を行う。

## 〔従来例2〕

次に、図5（a）に示すように、電子制御装置の動作を制御するマイコンとして、プログラムに従い動作するCPU（即ち、命令解読部や演算部等からなる中央演算装置）203とは別に、そのCPU203を停止状態から動作状態へと起床させるタイマブロック205を内蔵したマイコン201を用いる。

## 【0008】

つまり、このマイコン201において、CPU203は、実行すべき処理が無く、動作しなくても良い状態になったと判断すると、図5（b）に示すように、自分の動作を停止して、当該マイコン201の実質的な動作（即ち、プログラムを実行する動作）を停止させると共に、タイマブロック205へ動作指示を出す。

## 【0009】

すると、タイマブロック205は、計時を開始して、所定のタイマ時間が経過した時に、CPU203へタイムアップ信号を出力する。尚、タイマブロック205は、CPU203と共通の動作クロックであって、当該マイコン201内部の発振回路207で発生される高周波の動作クロックによって動作する。

## 【0010】

そして、CPU203は、タイマブロック205からのタイムアップ信号によって停止状態から動作状態へと起床し、プログラムの実行を再開して予定の動作に該当する処理を行う。

## 〔従来例3〕

次に、図6(a)に示すように、電子制御装置の動作を制御するマイコンとして、動作速度が可変のマイコン301を用いる。

## 【0011】

つまり、このマイコン301は、CPU303と、そのCPU303の動作クロックの元となる基準クロックを発生させる発振回路307と、その発振回路307で発生される基準クロックから、CPU303により指示される周波数のクロックを生成して、CPU303の動作クロックとして出力する周波数制御回路305とを備えている。

## 【0012】

そして、このマイコン301において、CPU303は、プログラムを実行している通常動作時には、周波数制御回路305から出力される動作クロックの周波数を通常の高い周波数（例えば数十MHz）に設定して、高速で動作するが、実行すべき処理が無く、動作しなくても良い状態になったと判断すると、図6(b)に示すように、周波数制御回路305から出力される動作クロックの周波数を通常よりも低い周波数（例えば数MHz）に設定して、低速で動作する（即ち、低消費電力状態になる）と共に、その状態で所定のタイマ時間の計時を行う。

## 【0013】

そして更に、CPU303は、その低速動作状態でタイマ時間が経過したと判断すると、周波数制御回路305から出力される動作クロックの周波数を再び通常の高い周波数に設定して、高速での動作を再開し、予定の動作に該当する処理を行う。

## 【0014】

## 【発明が解決しようとする課題】

ところが、まず、上記従来例1では、マイコン101とは別に専用のタイマ回路103が必要となるため、電子制御装置全体での回路規模とコストが増大してしまう。

## 【0015】

また、上記従来例2では、マイコン201の中核部であるCPU203は動作



を停止するものの、CPU 203を動作させるための高周波数のクロックを発生させる発振回路（即ち、CPUの動作クロックを発生する部分）207が常時動作することとなるため、消費電流を大幅に低減することができない。しかも、従来例2では、例えば数十時間や数日間といった極長時間のタイマを構成するのが非常に難しい。これは、専用のタイマブロック205で計時できる時間が有限なためであり、このことは従来例1でも同様である。

#### 【0016】

そして更に、上記従来例3では、発振回路307及び周波数制御回路305だけでなく、CPU303が常時動作することとなるため、消費電流の低減効果が小さい。

本発明は、こうした問題に鑑みなされたものであり、予定の動作を所定時間が経過してから実施する、というタイマ機能を、より少ない消費電力で実現するのに好適なマイクロコンピュータを提供することを目的としている。

#### 【0017】

##### 【課題を解決するための手段及び発明の効果】

上記目的を達成するためになされた請求項1に記載のマイコン（マイクロコンピュータ）は、プログラムに従い動作するCPUと、CPUを動作させるためのメインクロック（即ち、CPUの動作クロック）を生成するメインクロック生成手段と、上記メインクロックよりも周波数が低いサブクロックを受けて動作すると共に、CPUを間欠的に動作させるための制御を行う間欠動作制御手段とを備えている。

#### 【0018】

そして、このマイコンでは、「CPUが、動作を停止する際に間欠動作制御手段へ停止指令を出力し、間欠動作制御手段が、CPUからの停止指令を受けると、メインクロック生成手段の動作を停止させると共に、所定の設定時間の計時を開始し、その設定時間の経過後に、メインクロック生成手段の動作を再開させてCPUを停止状態から動作状態へと起床させる」という動作が繰り返されることにより、図3（A）、（B）の上段に示すようなCPUの間欠動作が実施される。

## 【 0 0 1 9 】

尚、CPUが停止状態となっている期間中（図3において「間欠時間」と記している期間中）は、当該マイコンの実質的な動作であるプログラムの実行動作が停止されると共に、CPUの動作クロックである高周波数のメインクロックも停止することとなる。また、間欠動作制御手段が計時する上記設定時間は、一般には、CPUによってセットされる（即ち、プログラマブルである）ように構成されるが、固定値であっても良い。

## 【 0 0 2 0 】

ここで特に、請求項1のマイコンは、サブクロックを受けて動作する間欠時間計測手段を備えており、その間欠時間計測手段は、図3（A）の下段に示すように、CPUが停止状態である期間（以下、間欠時間ともいう）を自動的に計測すると共に、その計測値がCPUによって読み取り可能に構成されている。

## 【 0 0 2 1 】

よって、このマイコンによれば、CPUが下記の如く動作するようにプログラムを設定すれば、たとえ数十時間や数日間といった非常に長い時間でも計測することができると共に、その時間計測機能（延いてはタイマ機能）を、非常に少ない消費電力で実現することができる。

## 【 0 0 2 2 】

即ち、図3（A）に示すように、CPUは、時間を計測する際に、間欠動作すると共に、停止状態から起床して一時動作する毎に、今回起床するまでの間欠時間 $T_a$ の値を間欠時間計測手段から読み取り、また、今回起床してからの動作時間（即ち、当該CPUが動作状態となっている時間） $T_b$ を自らソフト処理で計測する。そして更に、CPUは、停止状態から起床して動作する毎に、それまでの間欠回数分の上記 $T_a$ 及び $T_b$ の累積加算値（ $\Sigma(T_a + T_b)$ ）を求めて、その累積加算値を、最初に動作を停止した間欠動作開始時からの延べ時間として、所定時間が経過したか否かの判定処理に用いる。尚、CPUは、上記判定処理で所定時間が経過したと判断したならば、間欠動作を止めて、予定の動作に該当する処理を行う。

## 【 0 0 2 3 】

このような請求項1のマイコンによれば、CPUだけでなく該CPUの動作クロックである高周波数のメインクロックが停止した状態で、時間の計測が遂行されるため、予定の動作を所定時間が経過してから実施する、というタイマ機能を、より少ない消費電力で実現することができ、しかも、計測すべき所定時間（即ち、タイマ時間）が数十時間や数日間といった具合に非常に長くても、確実に対応することができる。つまり、間欠時間計測手段が連続して計測可能な時間は有限であるが、図3（A）に示したように、間欠時間計測手段により計測されるCPUの1回ずつの間欠時間 $T_a$ と、CPU自身のソフト処理により計測される該CPUの動作時間 $T_b$ との累積加算値（ $\Sigma(T_a + T_b)$ ）によって、時間の計測を行うことができるからである。

## 【0024】

また、請求項1のマイコンによれば、タイマ時間（即ち、経過したことを判断する所定時間）をソフトウェアによって自由に設定することができ、汎用性を損なうこともない。

そして更に、請求項1のマイコンによれば、図3（A）に示したように、CPUを定期的に動作させながら、継続して長い時間を計測することができるため、時間の計測中に処理を適宜変更することも可能になる。

## 【0025】

例えば、タイマ時間の計測を開始してから状況が変化して、実施予定の動作が不要になった場合には、そのことを検知して他の処理を実行することができるようになる。また、定期的に内部データ等をチェックして当該マイコンの動作の安定性を確認することができ、動作の信頼性を向上させることができる。

## 【0026】

次に、請求項2に記載のマイコンでは、請求項1のマイコンにおいて、間欠時間計測手段は、CPUからの動作モード切替指令によって、CPUが停止状態である期間を計測する第1の動作モードと、時間を継続して計測する第2の動作モードとの何れかの動作モードに設定されると共に、少なくとも、動作モードが第2の動作モードに設定されている場合には、CPUからのクリア指令によって計測値のクリアが行えるように構成されている。

## 【 0 0 2 7 】

このような請求項 2 のマイコンによれば、間欠時間計測手段の動作モードを第 1 の動作モードに設定すれば、前述した請求項 1 のマイコンと同じ使用方法で同じ効果が達成できる。

その上、この請求項 2 のマイコンによれば、CPU が間欠時間計測手段の動作モードを第 2 の動作モードに設定して下記の如く動作するように、プログラムを設定すれば、請求項 1 のマイコンについて述べた効果を、CPU の処理負荷を一層軽くしながらも実現することができる。

## 【 0 0 2 8 】

即ち CPU は、時間の計測を開始する時までに、間欠時間計測手段の動作モードを第 2 の動作モードに設定しておくと共に、図 3 (B) の時刻  $t_1$  に示すように、時間の計測を開始する時に、間欠時間計測手段へクリア指令を与えて、該間欠時間計測手段の計測値をクリアする (0 にする)。そして、図 3 (B) に示すように、CPU は、間欠動作を開始すると共に、停止状態から起床して一時動作する毎に、間欠時間計測手段の計測値  $T$  を読み取って、その直後に間欠時間計測手段の計測値をクリアする (つまり、間欠時間計測手段の計時動作を 0 からリスタートさせる)。そして更に、CPU は、停止状態から起床して動作する毎に、間欠時間計測手段から読み取った上記計測値  $T$  (図 3 (B) における  $T_1$ ,  $T_2$ , ...) の累積加算値 ( $\Sigma (T)$ ) を求めて、その累積加算値を、時間の計測を開始した時点からの延べ時間として、所定時間が経過したか否かの判定処理に用いる。尚、CPU は、上記判定処理で所定時間が経過したと判断したならば、間欠動作を止めて、予定の動作に該当する処理を行う。

## 【 0 0 2 9 】

そして、請求項 2 のマイコンを、このように使用すれば、前述した請求項 1 のマイコンと全く同じ効果を得ることができる上に、図 3 (A) の場合のように、CPU が動作状態となっている時間  $T_b$  を自らソフト処理で計測する必要がなくなるため、CPU の処理負荷を低減することができるのである。

## 【 0 0 3 0 】

## 【発明の実施の形態】

以下、本発明が適用された実施形態のシングルチップマイコンについて、図面を用いて説明する。

まず図 1 は、本実施形態のマイコン 1 の構成を表すブロック図である。

#### 【0031】

図 1 に示すように、本実施形態のマイコン 1 は、プログラムに従い動作する CPU 3、プログラムや固定データが予め格納される ROM 5、及び CPU 3 による演算結果を一時記憶するための RAM 7 からなる基本的な構成に加えて、当該マイコン 1 の外部に設けられる発振素子 8 と協同して CPU 3 の動作クロックであるメインクロック（本実施形態では数 MHz ～ 数十 MHz）を生成するメイン発振回路 9 及び該メイン発振回路 9 を制御する発振制御部 11 からなるメインクロック発生部 13 と、上記発振制御部 11 と協同して、CPU 3 を間欠的に動作（間欠動作）させるための制御を行う間欠動作制御部 15 と、CPU 3 の間欠時間（即ち、CPU 3 が間欠動作している際に停止状態である期間）を計測するタイマブロック 17 と、当該マイコン 1 の外部に設けられる発振素子 18 と協同して上記メインクロックよりも周波数が低いサブクロック（本実施形態では数十 KHz）を生成するサブ発振回路 19 とを備えている。

#### 【0032】

そして、本実施形態のマイコン 1 において、上記発振制御部 11 と、間欠動作制御部 15 と、タイマブロック 17 との各々は、サブ発振回路 19 で常時生成されるサブクロックを受けて動作する。

尚、本実施形態では、メイン発振回路 9 が、メインクロック生成手段に相当し、発振制御部 11 と間欠動作制御部 15 とが、間欠動作制御手段に相当し、タイマブロック 17 が、間欠時間計測手段に相当している。

#### 【0033】

ここで、CPU 3 は、前述した従来例 2 の CPU 203 と同様に、特定の動作停止命令を実行することによって自己の動作を停止することができるようになっている。そして、CPU 3 は、自ら動作を停止する時に（つまり、上記動作停止命令の実行時に）、間欠動作制御部 15 へ停止指令を出力するようになっている。

## 【 0 0 3 4 】

一方、間欠動作制御部 1 5 は、計時すべき時間（所定の設定時間に相当）が CPU 3 によってセットされる（書き込まれる）レジスタ 1 5 a を備えている。

そして、間欠動作制御部 1 5 は、通常時には、メインクロック発生部 1 3 の発振制御部 1 1 に動作指示を与えて、該発振制御部 1 1 にメイン発振回路 9 を動作させているが、CPU 3 からの上記停止指令を受けると（即ち、CPU 3 が動作を停止すると）、発振制御部 1 1 に停止指示を出力して、該発振制御部 1 1 にメイン発振回路 9 の動作を停止させると共に、上記レジスタ 1 5 a にセットされている時間（以下、設定時間という）の計時を開始し、その設定時間が経過すると、発振制御部 1 1 に再び動作指示を出力して、該発振制御部 1 1 にメイン発振回路 9 の動作を再開させる。そして更に、間欠動作制御部 1 5 は、CPU 3 から上記停止指令を受けて、発振制御部 1 1 へ停止指示を出力した時に、タイマブロック 1 7 へ、CPU 3 の動作が停止したことを示す停止報知信号を出力する。

## 【 0 0 3 5 】

尚、上記設定時間は、サブクロックの数（即ち周期数）に基づいて計時される。また、本実施形態において、上記停止報知信号は、極短いパルス幅のワンショットパルス信号である。

また、メインクロック発生部 1 3 の発振制御部 1 1 は、上記間欠動作制御部 1 5 からの動作指示と停止指示とに応じて、メイン発振回路 9 の動作と非動作とを切り替えるが、特に、間欠動作制御部 1 5 からの動作指示を受けてメイン発振回路 9 の動作を開始させた際には、その時点からメインクロックの周波数が安定すると見なされる所定の発振安定待ち時間が経過した時に、CPU 3 へ、該 CPU 3 を停止状態から動作状態へと起床させるための RUN 信号を出力する。

## 【 0 0 3 6 】

尚、上記発振安定待ち時間は、メインクロックの周波数が確実に安定してから CPU 3 を起床させるために設けられており、サブクロックの数に基づいて計時される。そして、発振制御部 1 1 は、上記発振安定待ち時間が CPU 3 によってセットされる（書き込まれる）レジスタ 1 1 a を備えている。また、本実施形態において、発振制御部 1 1 は、上記 RUN 信号を、間欠動作制御部 1 5 から次に

停止指示を受けるまで（つまり、メイン発振回路 9 の動作を次に停止させる時まで）継続して出力するようになっている。そして、その RUN 信号は、タイマブロック 1 7 にも供給されるようになっている。

#### 【 0 0 3 7 】

次に、タイマブロック 1 7 は、CPU 3 の間欠動作と連動して該 CPU 3 の間欠時間を自動的に計測する基本動作モードとしての第 1 の動作モードと、時間を継続して計測するフリーランの第 2 の動作モードとの、2 つの動作モードを有している。そして、タイマブロック 1 7 は、CPU 3 からの動作モード切替指令によって、動作モードが上記 2 つの動作モードの内の何れかに設定されると共に、時間の計測値が CPU 3 によって読み取り可能に構成されている。

#### 【 0 0 3 8 】

そして更に、タイマブロック 1 7 は、動作モードが第 1 の動作モードに設定されている場合には、間欠動作制御部 1 5 からの上記停止報知信号によって計測値が 0 にクリアされると共に、発振制御部 1 1 から上記 RUN 信号が出力されている間は、計時動作を停止して計測値を保持するように構成されている。

#### 【 0 0 3 9 】

また、タイマブロック 1 7 は、CPU 3 からのクリア指令によって、計測値のクリアが行われると共に、CPU 3 からのカウント停止指令によって、計時動作の停止及び計測値の保持が行われるように構成されている。

具体的に説明すると、図 2 に示すように、タイマブロック 1 7 は、時間を計測する手段として、サブクロックを受けてアップカウント動作すると共に、そのカウント値（時間の計測値に相当）が、CPU 3 に読み取られるカウンタ 2 1 を備えている。そして、カウンタ 2 1 は、リセット端子 2 1 a にハイレベルの信号が供給されると、カウント値が 0 にクリアされ、また、動作停止端子 2 1 b にハイレベルの信号が供給されている間、計時動作に相当するカウントアップ動作を停止してカウント値を保持するように構成されている。

#### 【 0 0 4 0 】

そして更に、タイマブロック 1 7 は、CPU 3 からの動作モード切替指令に該当する信号（以下、動作モード切替信号という）と間欠動作制御部 1 5 からの停

止報知信号との論理積信号を出力するアンド回路 2 3 と、CPU 3 からの動作モード切替信号と発振制御部 1 1 からの RUN 信号との論理積信号を出力するアンド回路 2 5 と、アンド回路 2 3 の出力と CPU 3 からのクリア指令に該当する信号（以下、クリア信号という）との論理和信号を、カウンタ 2 1 の上記リセット端子 2 1 a に供給するオア回路 2 7 と、アンド回路 2 5 の出力と CPU 3 からのカウント停止指令に該当する信号（以下、カウント停止信号という）との論理和信号を、カウンタ 2 1 の上記動作停止端子 2 1 b に供給するオア回路 2 9 とを備えている。

## 【 0 0 4 1 】

尚、本実施形態において、タイマブロック 1 7 に入力される各信号は、ハイレベルがアクティブレベルである。また、CPU 3 からオア回路 2 7 へのクリア信号は、間欠動作制御部 1 5 からの停止報知信号と同様に、極短いパルス幅のワンショットパルス信号である。

## 【 0 0 4 2 】

このようなタイマブロック 1 7 では、CPU 3 からの動作モード切替信号がハイレベル（＝論理 1）の場合に、間欠動作制御部 1 5 からの停止報知信号が、アンド回路 2 3 及びオア回路 2 7 を介してカウンタ 2 1 のリセット端子 2 1 a に供給されると共に、発振制御部 1 1 からの RUN 信号が、アンド回路 2 5 及びオア回路 2 9 を介してカウンタ 2 1 の動作停止端子 2 1 b に供給されるため、当該タイマブロック 1 7 の動作モードが第 1 の動作モードに設定される。

## 【 0 0 4 3 】

つまり、この場合には、図 3（A）に示すように、CPU 3 の動作が停止して間欠動作制御部 1 5 から上記停止報知信号が出力されると、カウンタ 2 1 のカウント値が 0 にクリアされて該カウンタ 2 1 のカウント動作が最初から開始され、その後、発振制御部 1 1 から RUN 信号が出力されて CPU 3 が動作を再開すると、再び CPU 3 の動作が停止して間欠動作制御部 1 5 から停止報知信号が出力されるまで、カウンタ 2 1 のカウント動作が停止してカウント値が保持される、という動作が繰り返されることとなる。

## 【 0 0 4 4 】



また逆に、CPU 3からの動作モード切替信号がローレベル（＝論理0）の場合には、間欠動作制御部 1 5からの停止報知信号がカウンタ 2 1のリセット端子 2 1 aに供給されることと、発振制御部 1 1からのRUN信号がカウンタ 2 1の動作停止端子 2 1 bに供給されることとが、アンド回路 2 3， 2 5によって防止されるため、当該タイマブロック 1 7の動作モードが第2の動作モードに設定される。

## 【 0 0 4 5 】

つまり、この場合には、間欠動作制御部 1 5からの停止報知信号と、発振制御部 1 1からRUN信号とに拘わらず、カウンタ 2 1は、フリーランの状態でアップカウント動作を行うからである。

そして更に、タイマブロック 1 7では、動作モードが上記2つの動作モードの何れに設定されている場合でも、CPU 3からクリア信号が出力されると、カウンタ 2 1のカウント値が0にクリアされ、また、CPU 3からカウント停止信号が出力されている間は、カウンタ 2 1のカウント動作が停止して該カウンタ 2 1のカウント値が保持される。

## 【 0 0 4 6 】

以上のような本実施形態のマイコン 1では、CPU 3が、間欠動作制御部 1 5のレジスタ 1 5 aへ任意の時間をセットすると共に、実行すべき処理がなくて動作を停止しても良いと判断すると、上記動作停止命令を実行して、自己の動作を停止すると共に間欠動作制御部 1 5へ停止指令を出力する、といった具合にプログラムを設定すれば、図 3（A），（B）の上段に示すようなCPU 3の間欠動作が実現されることとなる。

## 【 0 0 4 7 】

即ち、CPU 3が、動作を停止しても良いと判断して、自らの動作を停止すると共に間欠動作制御部 1 5へ停止指令を出力すると、間欠動作制御部 1 5が、発振制御部 1 1に停止指示を出力してメイン発振回路 9の動作を停止させると共に、レジスタ 1 5 aにセットされている設定時間の計時を開始し、その設定時間が経過すると、発振制御部 1 1に再び動作指示を出力してメイン発振回路 9の動作を再開させることとなる。そして、その時点から、前述の発振安定待ち時間が経

過すると、発振制御部 1 1 から CPU 3 へ RUN 信号が出力されて、CPU 3 が停止状態から動作状態へと起床することとなり、以後は、こうした動作が繰り返されることにより、CPU 3 の間欠動作が実施される。

## 【 0 0 4 8 】

ここで特に、本実施形態のマイコン 1 では、サブクロックを受けて動作するタイマブロック 1 7 が設けられており、そのタイマブロック 1 7 は、動作モードが CPU 3 によって第 1 の動作モードに設定されている場合には、図 3 (A) を用いて説明したように、CPU 3 の実際の間欠時間（即ち、CPU 3 が動作を停止して間欠動作制御部 1 5 が停止報知信号を出力した時点から、発振制御部 1 1 が CPU 3 へ RUN 信号を出力するまでの時間）を自動的に計測する。

## 【 0 0 4 9 】

よって、本実施形態のマイコン 1 によれば、CPU 3 がタイマブロック 1 7 の動作モードを第 1 の動作モードに設定して下記の如く動作するように、プログラムを設定すれば、たとえ数十時間や数日間といった非常に長い時間でも計測することができると共に、その時間計測機能（延いてはタイマ機能）を、非常に少ない消費電力で実現することができる。

## 【 0 0 5 0 】

即ち、図 3 (A) に示すように、CPU 3 は、時間を計測する際に、間欠動作すると共に、停止状態から起床して一時動作する毎に、今回起床するまでの間欠時間  $T_a$  に相当するカウント値をタイマブロック 1 7 のカウンタ 2 1 から読み取って、そのカウント値に基づき今回の実際の間欠時間  $T_a$  を算出し、また、今回起床してからの動作時間（即ち、当該 CPU 3 が動作状態となっている時間） $T_b$  を自らソフト処理で計測する。そして更に、CPU 3 は、停止状態から起床して動作する毎に、それまでの間欠回数分の上記間欠時間  $T_a$  及び動作時間  $T_b$  の累積加算値（ $\Sigma (T_a + T_b)$ ）を求めて、その累積加算値を、最初に動作を停止した間欠動作開始時からの延べ時間として、所定時間が経過したか否かの判定処理に用いる。そして、CPU 3 は、上記判定処理で所定時間が経過したと判断したならば、間欠動作を止めて、予定の動作に該当する処理を行う。

## 【 0 0 5 1 】

このようなマイコン 1 によれば、CPU 3 だけでなく該 CPU 3 の動作クロックである高周波数のメインクロックが停止した状態で、時間の計測が遂行されるため、予定の動作を所定時間が経過してから実施する、というタイマ機能を、より少ない消費電力で実現することができ、しかも、計測すべき所定時間（タイマ時間）が数十時間や数日間といった具合に非常に長くても、確実に対応することができる。つまり、タイマブロック 17 が連続して計測可能な時間は有限であるが、図 3（A）に示したように、タイマブロック 17 により計測される CPU 3 の 1 回ずつの間欠時間  $T_a$  と、CPU 3 自身のソフト処理により計測される該 CPU 3 の動作時間  $T_b$  との累積加算値（ $\Sigma (T_a + T_b)$ ）によって、タイマ時間の計測を行うことができるからである。

## 【0052】

また、本実施形態のマイコン 1 によれば、経過したことを判断するタイマ時間をソフトウェアによって自由に設定することができ、汎用性を損なうこともない。

そして更に、本実施形態のマイコン 1 によれば、図 3（A）に示したように、CPU 3 を定期的に動作させながら、継続して長い時間を計測することができるため、時間の計測中に処理を適宜変更することも可能になる。例えば、タイマ時間の計測を開始してから状況が変化して、実施予定の動作が不要になった場合には、そのことを検知して他の処理を実行することができ、また、定期的に RAM 7 内のデータ等をチェックして当該マイコン 1 の動作の安定性を確認することができ、動作の信頼性を向上させることができる。

## 【0053】

また、本実施形態のマイコン 1 において、タイマブロック 17 は、CPU 3 からの動作モード切替指令によって、時間を継続して計測する第 2 の動作モードに設定できると共に、少なくともその第 2 の動作モードに設定されている場合には、CPU 3 からの指令によってカウント値のクリアと保持とが行えるようになっている。

## 【0054】

このため本実施形態のマイコン 1 によれば、CPU 3 がタイマブロック 17 の

動作モードを第2の動作モードに設定して下記の如く動作するように、プログラムを設定すれば、前述の各効果を、CPU3の処理負荷を一層軽くしながらも実現することができる。

#### 【0055】

即ち CPU3は、時間の計測を開始する時までに、タイマブロック17の動作モードを第2の動作モードに設定しておくと共に、図3（B）の時刻 $t_1$ に示すように、時間の計測を開始する時に、タイマブロック17へクリア信号（クリア指令）を与えて、カウンタ21のカウント値を0にクリアする。そして、図3（B）に示すように、CPU3は、間欠動作を開始すると共に、停止状態から起床して動作を再開する毎に、タイマブロック17のカウンタ21のカウントアップ動作を一時停止させて該カウンタ21からカウント値を読み取ると共に、その直後にカウンタ21のカウント値をクリアして、該カウンタ21のカウント動作を0からリスタートさせる。そして更に、CPU3は、停止状態から起床して動作を再開する毎に、上記カウンタ21から読み取ったカウント値の累積加算値（即ち、図3（B）における $T_1$ ,  $T_2$ , …の累積加算値（ $\Sigma(T)$ ）に相当する値）を求めて、その累積加算値を、時間の計測を開始した時点からの延べ時間に相当する値として、所定時間が経過したか否かの判定処理に用いる。尚、CPU3は、上記判定処理で所定時間が経過したと判断したならば、間欠動作を止めて、予定の動作に該当する処理を行う。

#### 【0056】

そして、本実施形態のマイコン1を、このように使用すれば、図3（A）の場合のように、CPU3が動作状態となっている時間 $T_b$ を自らソフト処理で計測する必要がなくなるため、CPU3の処理負荷を低減することができるのである。

#### 【0057】

以上、本発明の一実施形態について説明したが、本発明は、種々の形態を採り得ることは言うまでもない。

例えば、間欠動作制御部15が計時する時間は、CPU3によってセットされる（プログラマブルである）のではなく、固定値であっても良い。但し、その時

間を任意に設定できる上記実施形態の構成を採用した方が、汎用性が高く有利である。

【図面の簡単な説明】

- 【図 1】 実施形態のマイコンの構成を表すブロック図である。
- 【図 2】 タイマブロックの構成を説明するブロック図である。
- 【図 3】 実施形態のマイコンの使用方法及び作用を説明する説明図である。
- 【図 4】 従来例 1 を説明する説明図である。
- 【図 5】 従来例 2 を説明する説明図である。
- 【図 6】 従来例 3 を説明する説明図である。

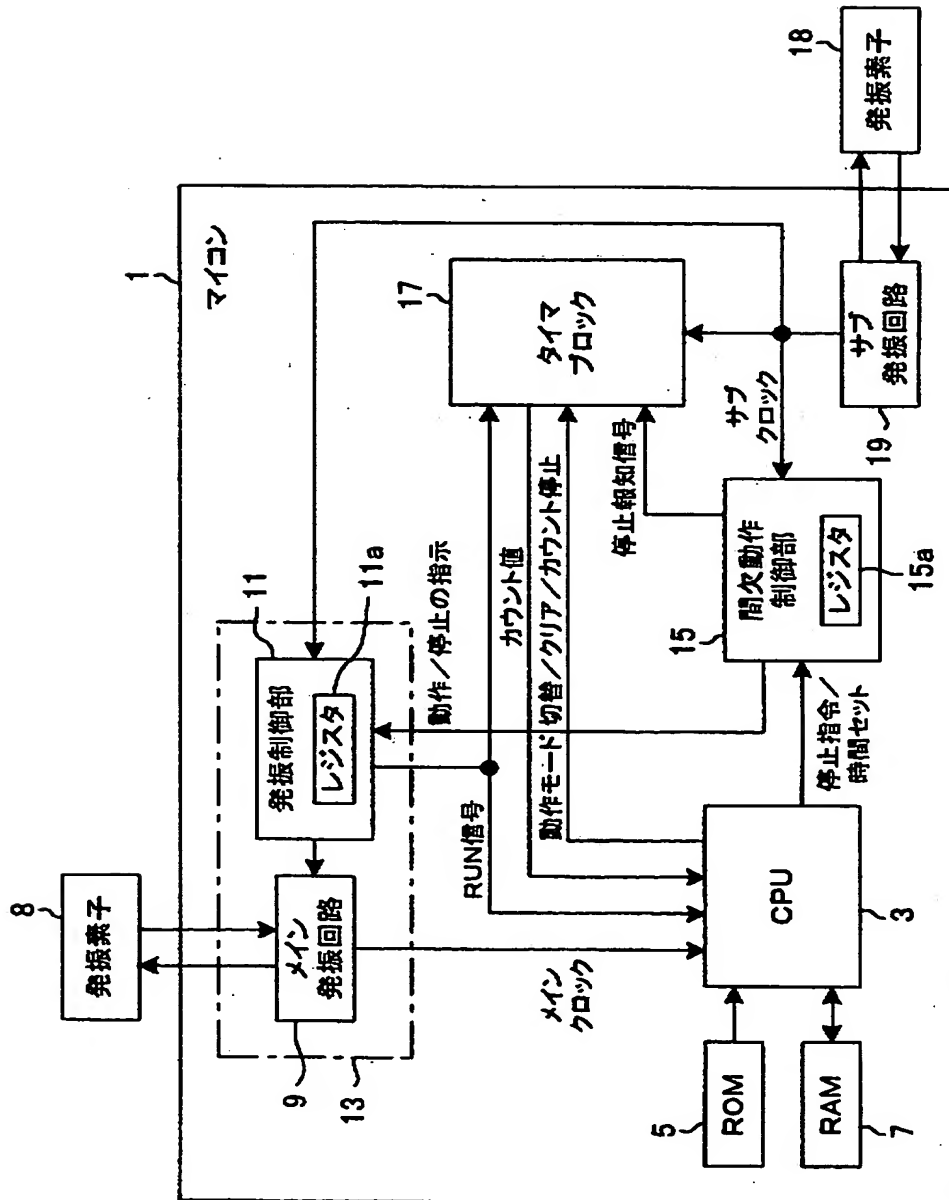
【符号の説明】

- 1 … マイコン (マイクロコンピュータ)      3 … CPU      5 … ROM
- 7 … RAM      8, 18 … 発振素子      9 … メイン発振回路
- 11 … 発振制御部      13 … メインクロック発生部      15 … 間欠動作制御部
- 11a, 15a … レジスタ      17 … タイマブロック      19 … サブ発振回路
- 21 … カウンタ      21a … リセット端子      21b … 動作停止端子
- 23, 25 … アンド回路      27, 29 … オア回路

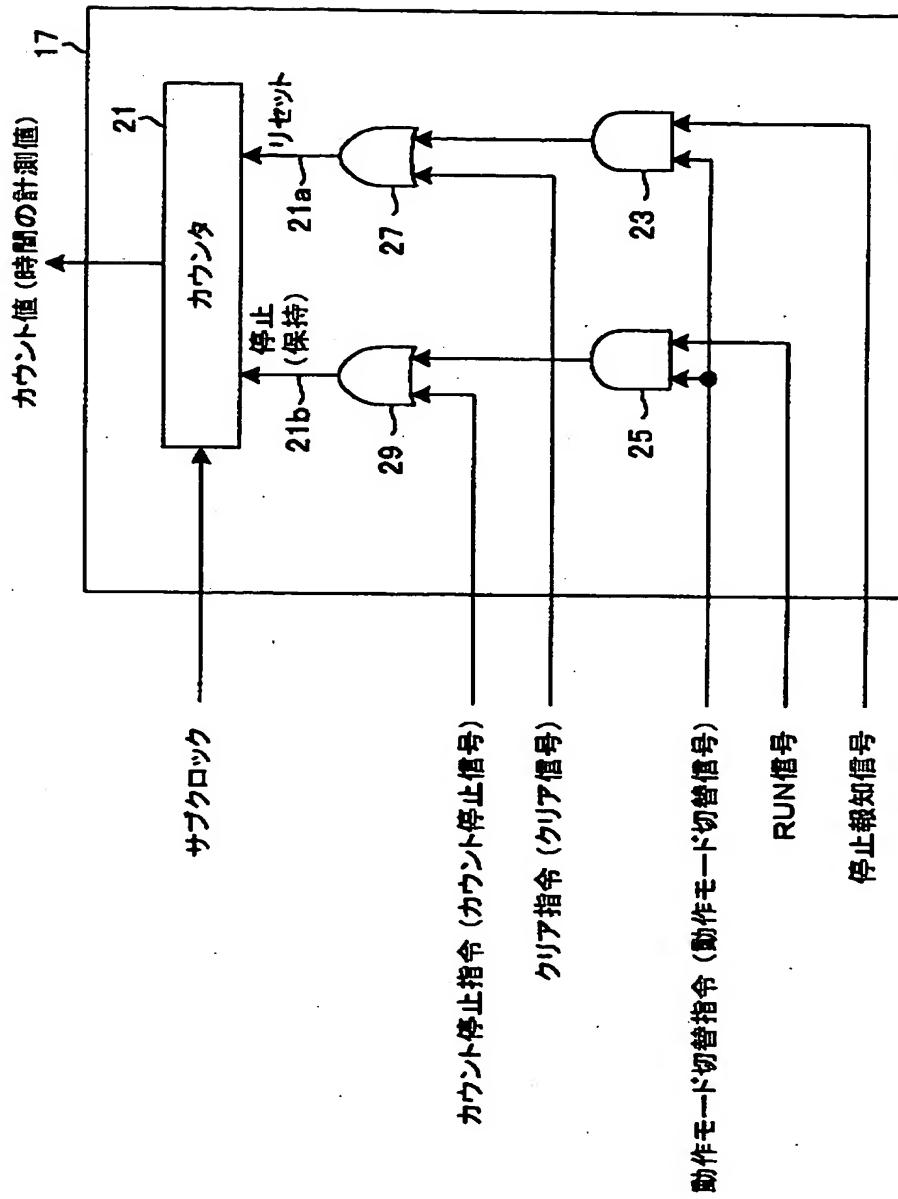
【書類名】

図面

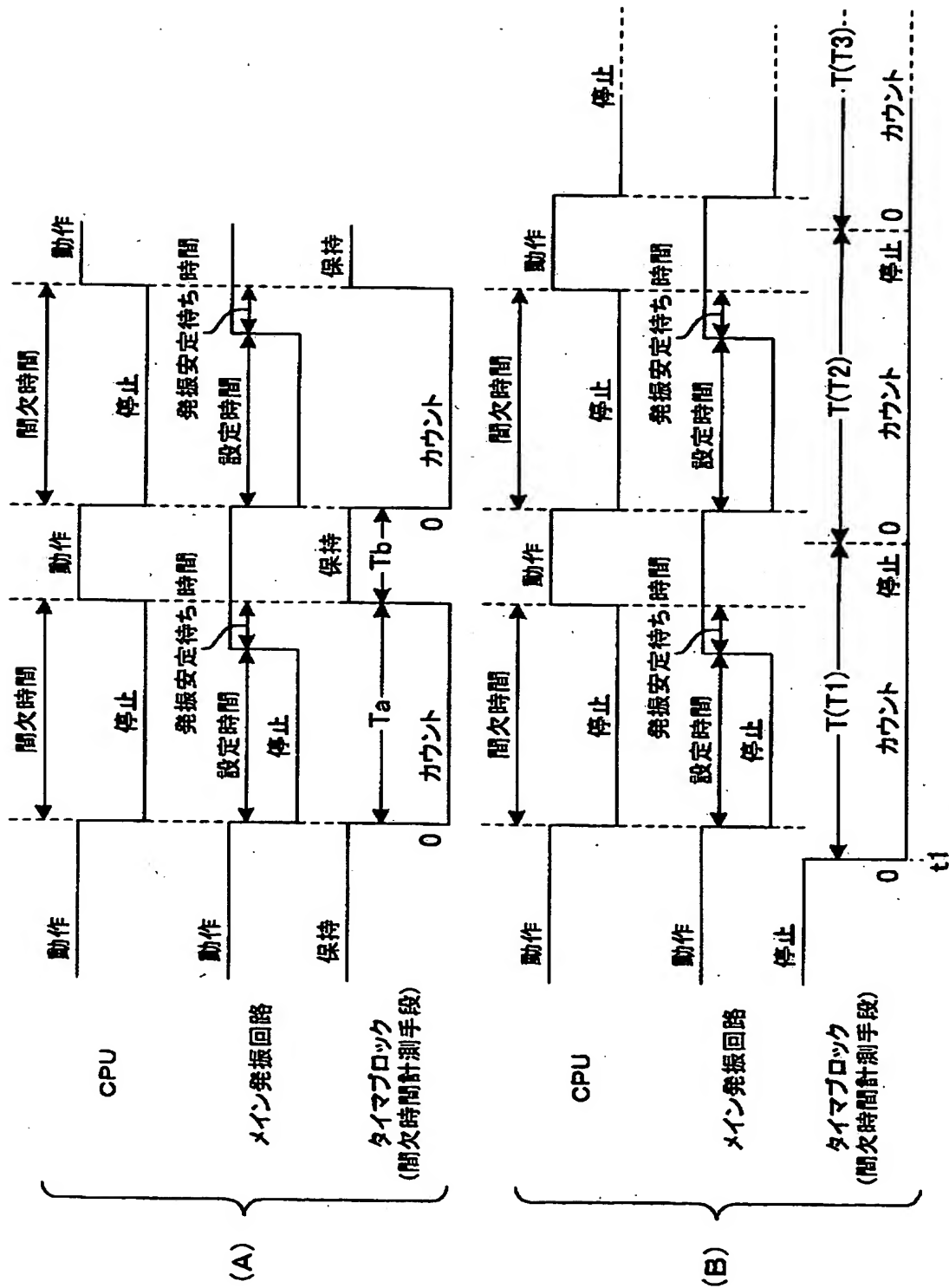
【図 1】



【図2】

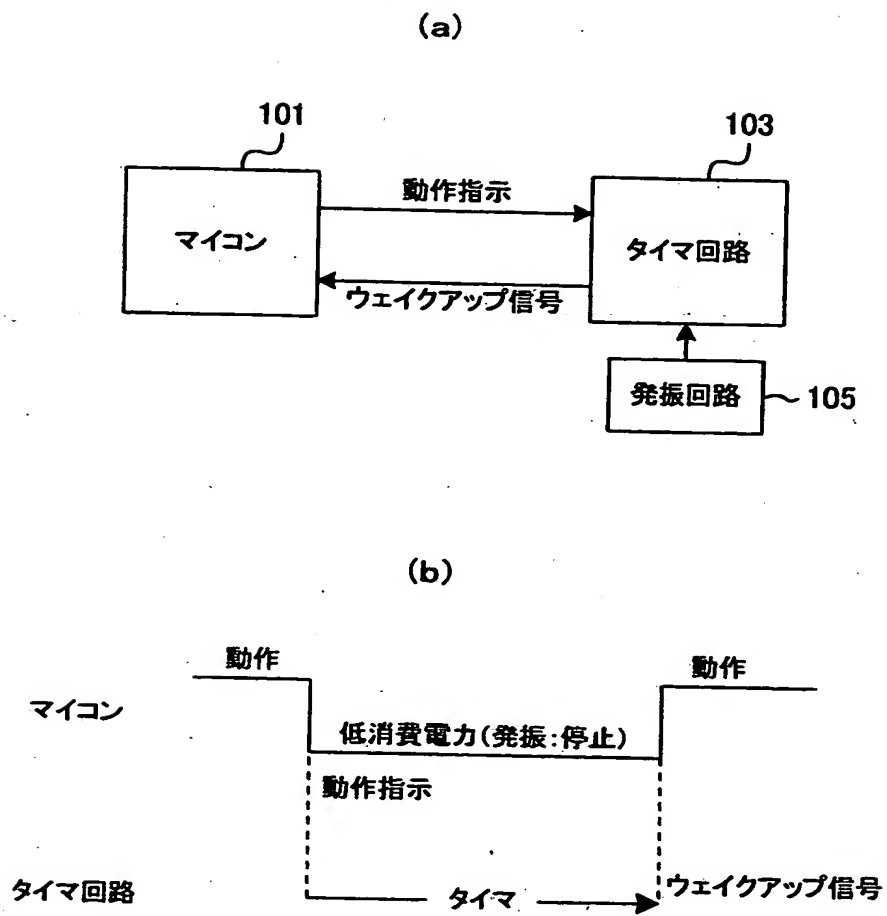


【図 3】

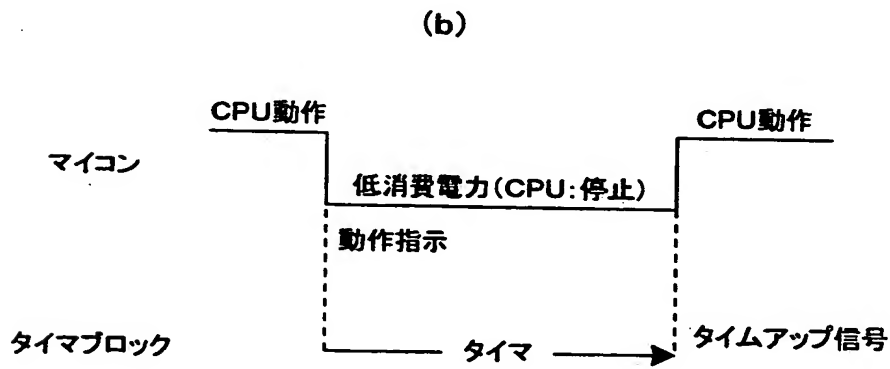
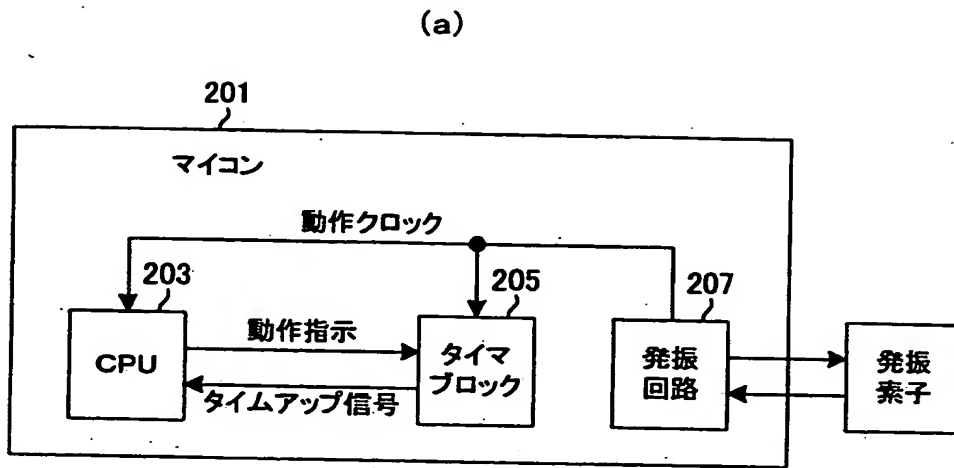




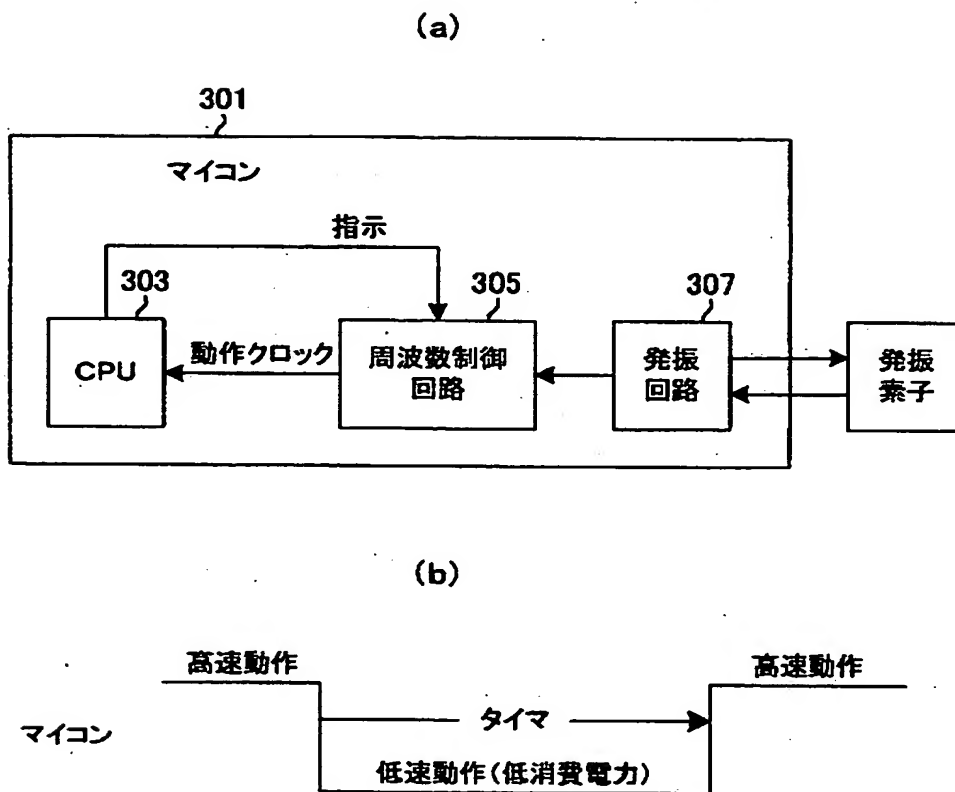
【図 4】



【図 5】



【図 6】



【書類名】 要約書

【要約】

【課題】 タイマ機能を低消費電力で実現するのに好適なマイコンを提供する。

【解決手段】 CPU3と、CPU3を動作させるためのメインクロックを生成する発振回路9と、低周波のサブクロックを受けてCPU3及び発振回路9を間欠的に動作させるための制御を行う間欠動作制御部15及び発振制御部11とを備えたマイコン1には、上記サブクロックを受けて動作して、CPU3が停止している実際の間欠時間を自動的に計測するタイマブロック17が設けられている。このマイコン1によれば、時間を計測する際に、CPU3が、間欠動作すると共に、停止状態から起床する毎に、今回起床するまでの間欠時間 $T_a$ をタイマブロック17から読み取り、また、今回起床してからの当該CPU3の動作時間 $T_b$ を自らソフト処理で計測し、更にそれまでの上記 $T_a$ 及び $T_b$ の累積加算値を求めるようにすれば、極長い時間でも非常に少ない消費電力で計測可能となる。

【選択図】 図1

出 願 人 履 歴 情 報

識別番号 [000004260]

1. 変更年月日	1996年10月 8日
[変更理由]	名称変更
住 所	愛知県刈谷市昭和町1丁目1番地
氏 名	株式会社デンソー